REMARKS

Claims 1-22, 27-28, and 31 are cancelled, claims 23, 30, and 32 are amended, and claims 23-26, 29-30, and 32 are pending.

The Examiner is thanked for the indication that Claims 27 and 28 are directed to allowable subject matter. The independent claims 23, 30, and 32 have been amended to include the subject matter of Claim 27, and claims 27 and 28 have been cancelled. The remaining pending claims are believed to be allowable for at least the reasons that Claims 23, 30, and 32 are allowable.

The Office Action indicated one of the references listed on the PTO-SB-08 (GRAMANN; ABF Algorithms Implemented at ARL:UT; ARL-TR-92-7; U. of Texas Appl. Res. Lab., May 1992) was not listed on the parent patent application (6,724,916), and therefore requested that Applicants provide a copy of the reference. A copy is enclosed as an attachment to this paper. However, please note that this document appears to be listed as the last reference on the first column of U.S. Patent No. 6,724,916. Applicants request that the Examiner make this reference of record in the application by initialing and returning the earlier submitted PTO-SB-08 listing this reference. No fee is believed to be due under 37 CFR 1.97 for this submission, however, should such a fee be required, kindly charge the fee or credit overpayment to Deposit Account 50-0281.

In view of the above amendment, applicant believes the pending application is in condition for allowance. An early indication of the allowability of the application in the form of a Notice of Allowance is earnestly solicited.

5150

ARL-TR-92-7

Copy No. 24

ABF Algorithms Implemented at ARL:UT

Technical Report under Contract N00039-91-C-0082,
TD No. 01A1002, FDS System Engineering and Acoustics

Richard A. Gramann

Applied Research Laboratories
The University of Texas at Austin
P. O. Box 8029 Austin, TX 78713-8029

12 May 1992

Technical Report

Approved for public release; distribution is unlimited.

Prepared for:

Space and Naval Warfare Systems Command
Department of the Navy
Washington, D.C. 20363-5100

1. INTRODUCTION

1.1 BACKGROUND

Adaptive beamforming (ABF) has been in use on several of the Fleet sonar arrays for many years. It has also proven to be a useful tool to the Sound Surveillance Underwater System (SOSUS) community, enhancing sonar performance through the reduction of overall beam noise, interference rejection, and consequently enhanced detection range. Thus, ABF has become an important asset in processing line array data for passive detection problems.

It is anticipated that ABF will play a significant role in future U.S. Navy acoustic systems. Consequently, ARL:UT has implemented a variety of ABF algorithms. Several ABF algorithms have been implemented on the ARL:UT Alliant mini-supercomputer so that comparisons between these algorithms can be made on the same data, and the advantages (or disadvantages) of the different implementations explored. This document describes two of the ABF algorithms implemented, and some of the methods employed in these implementations. A previous document[1] describes much of the processing performed after the ABF weights have been calculated. The reader is referred to this reference for those details.

2. ABF ALGORITHMS

2.1 GENERAL DESCRIPTION

The ABF algorithms currently implemented at ARL:UT can be classified as element-level adaptive beamformers, in which the signal from each sensor is multiplied by a complex weight, and the weighted outputs summed to form the array output. The data provided to the beamformer is a cross spectral matrix (CSM) that spans a specified frequency range, for a particular set of acoustic sensors. Thus, the data have been preprocessed into the frequency domain prior to being read into the beamforming programs. The weights for each particular sensor are calculated based on this frequency domain representation of the data. Thus each sensor has one complex weight for each frequency, and the output of the array is then defined in the frequency domain as well.

The ABF algorithms described in this report can be divided into two types. The first algorithm has a single linear constraint, applied in the look direction, to assure that the array has unity gain in that direction. The second algorithm has other linear constraints in addition to the look direction constraint. The first ABF is referred to as the "single point constraint" (SP-ABF) algorithm; the second is referred to as the "multiple linear constraint" (MLC-ABF) algorithm. The types of additional linear constraints include derivative constraints and point constraints. Both are implemented in the current software. Details on each of these constraints are given Section 2.4. In addition to the linear constraint, a white noise gain constraint (WNC) has been implemented in both algorithms. The WNC is described in Sections 2.3 and 2.4.

All of the ABF algorithms discussed in this report assume plane wave signals. The arrays are assumed to be linear (i.e., straight) and sufficiently small that speed of sound variations across the array are negligible. Each sensor is assumed to be a simple omni-directional type sensor with frequency response sufficient for the frequency ranges of interest. Further, it is assumed that the signal digitization and filtering are properly performed such that the frequency domain representations of the signals are accurate.

One final note concerning the ABF algorithms implemented at ARL:UT is necessary. We refer to these algorithms as adaptive beamforming algorithms. They are, however, the optimum beamforming solution for the type of beamformer (i.e., SP-ABF or MLC-ABF) chosen. This is referred to as an "estimate and plug" solution. The adaptive

(i.e., realtime) part of the algorithm has not been included in these implementations. In a realtime system, the weights adapt with each CSM realization, usually through a recursive algorithm. Consequently the solution depends not only on the latest CSM, but on previous CSMs and the solutions associated with them. The current implementations allow a more exact comparison between the algorithm solutions, and provide an estimate of the performance of the converged adaptive (i.e., realtime) system. Although most realtime systems use an adaptive implementation, the performance is expected to be close to that of the optimum solutions.

2.2 SINGLE POINT CONSTRAINT ABF

The single point constraint adaptive beamformer (SP-ABF) is described and derived below. This beamformer is not explicitly implemented on ARL:UT's Alliant computer. Rather, its solution can be obtained from the beamformer implemented with a white noise gain constraint. Hence, the Alliant implementation description is included in Section 2.3.2. However, the SP-ABF is a basic minimum variance beamformer, and its derivation shows the basic mathematical methods involved in the more complicated cases. Thus, its derivation is included here for completeness.

2.2.1 Description and Derivation

The SP-ABF algorithm is derived by representing the output of the array (i.e., beamformed output power) as

$$P_{ABF_{out}} = w^H R w \quad , \tag{2.1}$$

where w is the weight vector, H denotes the Hermitian (or complex conjugate) transpose, and R is the CSM. The output power of the beamformer, defined above, is minimized, subject to a single constraint. In this case, the look direction response of the array is constrained to unity gain. Restating the problem in mathematical terms gives

$$\min_{w} w^H R w \quad \text{subject to } w^H d = 1 \quad , \tag{2.2}$$

where d is the look direction steering vector.

To solve this problem, the method of Lagrange multipliers is used. Forming the functional that adjoins the constraint to the function to be minimized gives

$$F = w^H R w - \lambda(w^H d - 1) \quad . \tag{2.3}$$

Taking the derivative of the functional with respect to the weight vector and setting it to zero, one obtains

$$\frac{dF}{dw} = Rw - \lambda d = 0 \quad . \tag{2.4}$$

Rewriting gives

$$w = \lambda R^{-1} d \quad . \tag{2.5}$$

Since $w^H d = 1$, one can solve for λ, and write the final solution for the weight vector as

$$w = \frac{R^{-1} d}{d^H R^{-1} d} \quad . \tag{2.6}$$

This is the optimum beamformer solution subject to the unity gain look direction constraint. This beamformer is often referred to as a minimum variance distortionless response (MVDR) beamformer. Unfortunately, this beamformer solution is very sensitive to sources of mismatch.[2] Thus a more robust beamforming solution is generally needed.[3] The more robust beamformer incorporates a white noise gain constraint, and is described in the next section.

2.2.2 Implementation on the Alliant

The MVDR (without WNC) beamformer has not been explicitly implemented on the Alliant. Rather, if a user would like this solution, the white noise gain constraint can be set to a very low value (e.g., -100 dB), essentially disabling it. Additional details are provided below.

2.3 SINGLE POINT CONSTRAINT ABF WITH WHITE NOISE GAIN CONSTRAINT

2.3.1 Description and Derivation

The SP-ABF with a white noise gain constraint is derived in a similar manner to the optimum beamformer described in the previous section. The idea behind this solution is to make the beamformer more robust, or tolerant of errors. One means of doing this is to constrain the white noise gain of the system. The white noise gain, for the MVDR beamformer, is defined as

$$G_w = \frac{1}{w^H w} \quad . \tag{2.7}$$

The white noise gain is constrained to be a value greater than or equal to the white noise gain constraint (WNC), δ^2. By definition, the white noise gain must be less than the number of elements in the array. Thus, when constrained, the white noise gain will fall in the range

$$\delta^2 \le G_w \le N_{elem} \quad . \tag{2.8}$$

Thus the problem can be stated as

$$\min_w \; w^H R w \quad \text{subject to } w^H d = 1 \text{ and } G_w \ge \delta^2 \quad . \tag{2.9}$$

The white noise gain constraint is a quadratic inequality constraint, and thus the solution cannot be derived algebraically in closed form. The solution is the same as that of the previous optimum beamformer, except the CSM, represented by R, is replaced by a CSM that has had "white noise" injected into it. Thus, the CSM is represented by $(R+\varepsilon I)$, where ε (the second Lagrange multiplier) is increased to the extent needed to satisfy the WNC. The solution now takes the form

$$w = \frac{(R+\varepsilon I)^{-1} d}{d^H (R+\varepsilon I)^{-1} d} \quad . \tag{2.10}$$

Adding a small amount to the diagonal elements of the CSM (i.e., εI) can be thought of in several ways. It de-emphasizes the contributions of the off-diagonal elements, which decreases the relative correlation seen between those corresponding array elements. This makes the CSM appear to have weaker directional sources. Consequently the power minimization does not attempt to "null" these contributions as much as it normally would without the increased diagonal contribution. Also, by increasing the diagonal contribution, the CSM is becoming more like a CSM of only white noise (diagonal matrix). As ε approaches infinity, the CSM becomes diagonally dominant, and the resulting weights are the same as those for the conventional beamformer (CBF). This is the upper limit of the white noise gain.

2.3.2 Implementation on the Alliant

The implementation of the SP-ABF on the Alliant involves additional derivations and equations. First, the CSM can be written in terms of its eigenvalues (σ) and corresponding eigenvectors (U). Thus

$$R = U\Sigma U^H \text{ or } Udiag(\sigma)U^H \quad . \tag{2.11}$$

The CSM with white noise injected can be written similarly,

$$R + \varepsilon I = U(\Sigma + \varepsilon I)U^H \quad , \tag{2.12}$$

with the inverse being

$$(R + \varepsilon I)^{-1} = U(\Sigma + \varepsilon I)^{-1}U^H \quad . \tag{2.13}$$

The eigenvalues and eigenvectors are calculated using a singular value decomposition routine (SVD). Rewriting the weights solution in terms of the eigenvalues and eigenvectors gives

$$w = \frac{Udiag\left(\frac{1}{(\sigma+\varepsilon)}\right)U^H d}{d^H Udiag\left(\frac{1}{(\sigma+\varepsilon)}\right)U^H d} \quad . \tag{2.14}$$

Rewriting the expression for G_W in terms of the eigenvalues and eigenvectors gives

$$G_w = \frac{\left[d^H U diag\left(\frac{1}{(\sigma+\varepsilon)} \right) U^H d \right]^2}{\left[d^H U diag\left(\frac{1}{(\sigma+\varepsilon)^2} \right) U^H d \right]} \quad (2.15)$$

This equation can be broken into the part that needs to be calculated once, and the part that varies with the iteration on ε. Rewriting Eq. (2.15) gives

$$G_w = \frac{\left[\sum_k \left| U^H d \right|_k^2 \left(\frac{1}{(\sigma_k+\varepsilon)} \right) \right]^2}{\left[\sum_l \left| U^H d \right|_l^2 \left(\frac{1}{(\sigma_l+\varepsilon)} \right)^2 \right]} \quad (2.16)$$

This expression (2.16) is first evaluated with ε=0. If it satisfies the WNC, then no ε is added to the CSM. If not, this expression is evaluated iteratively to find the proper value of ε that satisfies the WNC. A version of Brent's method, adapted from *Numerical Recipes*, 4,* is used to find the value of ε that satisfies the white noise constraint. Since ε (as well as G_W), can span several orders of magnitude, ε is parameterized in terms of $\log(\varepsilon)$ within the searching subroutine. Once a value of ε is found that satisfies the constraint, the weights are evaluated using Eq. (2.14). The weights for one look direction are calculated for all frequencies and are written to disk, prior to calculating the next look direction.

The white noise gain constraint is set relative to the maximum white noise gain possible for the array in use. In the current implementations, for frequencies above the design frequency of the array, the maximum white noise gain value is 10 $\log(N_{elem})$, where N_{elem} is the number of sensors. This is the maximum gain attainable against spherically isotropic noise (i.e., the directivity index DI). At frequencies less than the design frequency of the array, the maximum white noise gain is set to 10 $\log(2L_a/\lambda)$, where L_a is defined as

* Function ZBRENT, pp. 253-254.

$$L_a = L\left(\frac{N_{elem}}{N_{elem}-1}\right) = (x_{max} - x_{min})\left(\frac{N_{elem}}{N_{elem}-1}\right) \qquad (2.17)$$

and x are the sensor locations, L is the actual distance between the end sensors (i.e., x_{max} and x_{min}), and λ is the wavelength at that frequency. These define the maximum value of white noise gain, and the WNC is set relative to these values. Figure 2.1 shows an example of the white noise gain limit and a constraint curve -3 dB relative to this maximum. The array has eight elements. Based on this constraint scheme, the white noise gain will fall on or between the two curves.

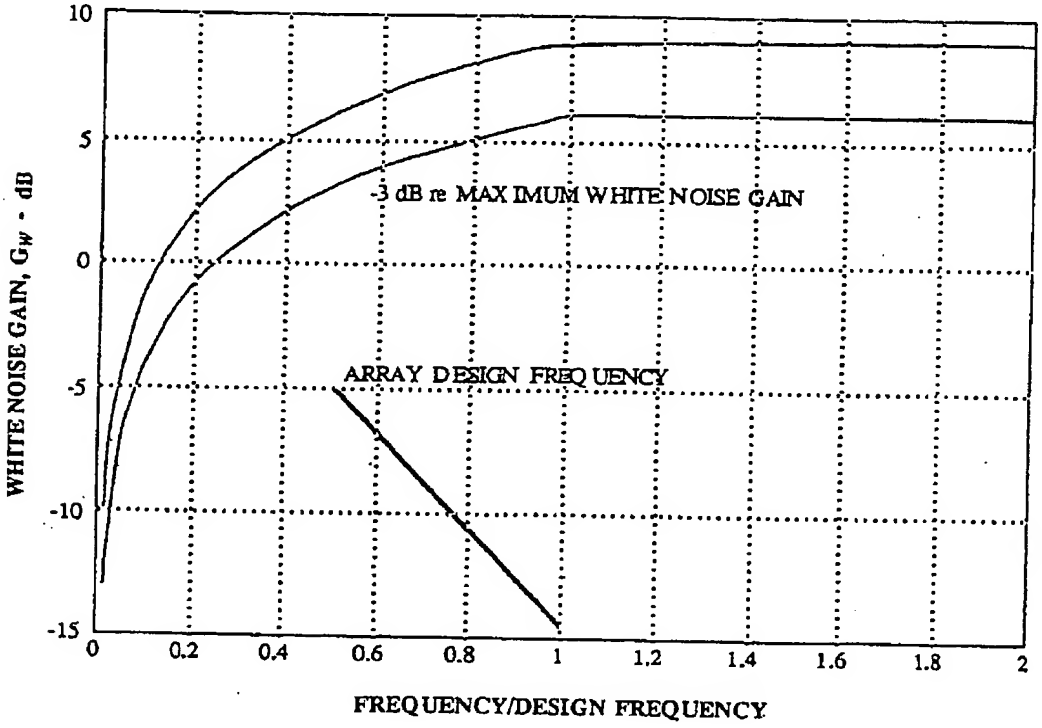

FIGURE 2.1
MAXIMUM WHITE NOISE GAIN AND EXAMPLE CONSTRAINT (8-ELEMENT ARRAY)

AS-92-114

Figure 2.1 shows the current white noise gain constraint implementation. This implementation is a compromise solution that is currently being used in the existing software. Several corrections to this constraint (e.g., setting the maximum white noise gain

limit to 0 dB for very large wavelengths, accounting for variations in steering direction, etc.) will be examined in future work, and reported in a later document.

2.4 MULTIPLE LINEAR CONSTRAINTS WITH WHITE NOISE GAIN CONSTRAINT

2.4.1 Description and Derivation

Multiple linear constraints can be added to the beamformer solution to provide additional control of the beamformer response. These additional constraints can be used to force the beam response curve to have its first, and possibly higher derivatives equal to zero in the look direction. This type of linear constraint is known as a derivative constraint. Thus one can set the first derivative to zero in the look direction, forcing the beam response curve (i.e., the beam pattern) to be relatively flat in the vicinity of the look direction.

Another type of linear constraint is the point constraint. This constraint is used to set a specified response at locations other than the look direction. Point constraints are generally used as mainlobe maintenance constraints. One example is the "three-point constraint," where the look direction and one point on either side of the look direction is set to a specific level. Another constraint scheme might be to match the CBF beam response at specific locations on either side of the look direction. This is referred to as the Match-CBF scheme. The user selects the points (in dB) at which they wish to have the ABF beam response match the CBF beam response. This defines the location (in sin θ space) that these constraints are applied. They are calculated for the specific array and the locations of the constraints vary with frequency. The current implementation on the Alliant allows the user to select the response points to match the CBF pattern, and then iteratively finds the locations of those constraint points. A "three point" scheme could be to match the CBF response where the CBF response is "3 dB down." Another possible "three point" scheme could be to match the CBF response at the main response axis (MRA) of the adjacent beams in a multi-beam system. This scheme has not been implemented to date, but could easily be incorporated in the existing code. One can also specify response levels at locations relative to the look direction, which are independent of frequency. These are strictly a "user defined" point constraint.

The multiple linear constraint ABF (MLC-ABF) is derived in the same fashion as the SP-ABF. In this case, however, the look direction constraint ($w^H d = 1$) is replaced with a constraint matrix and a response vector. Thus the single constraint becomes

$$M^H w = g, \tag{2.18}$$

where M is the constraint "location" matrix, w is the weights vector, and g is the response vector.

Derivative Constraints

M is constructed differently for derivative constraints than for point constraints. For derivative constraints, the first column of M is the steering vector for the look direction. Subsequent columns correspond to the derivatives of the steering vector (i.e., the second column is the 1st derivative, the third is the 2nd derivative, etc.), up to the number of derivatives being set to zero. Thus, the first column is

$$M_{k,1} = \exp\left(-j\frac{2\pi f x_k \sin(\theta)}{c}\right), \tag{2.19}$$

where f is the frequency, c is the speed of sound, and θ is the look direction. The second column contains the 1st derivative, and is calculated using the following expression:

$$M_{k,2} = \exp\left(-j\frac{2\pi f x_k}{c}\right) M_{1,k}. \tag{2.20}$$

Higher order derivative columns then become

$$M_{k,nderiv+1} = M_{k,2} M_{k,nderiv}. \tag{2.21}$$

The response vector associated with this constraint matrix has 1 for its first element, corresponding to the look direction response, and zeroes for all of the derivative terms. Thus, g takes the form

$$g = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdots \end{bmatrix} , \tag{2.22}$$

with as many zeroes as derivatives being set.

Point Constraints

The point constraint matrix is constructed with the columns forming the direction vectors for the constraint locations. Thus, the first column is the look direction, as before, with subsequent columns forming the look direction vectors for the additional point constraints. Hence,

$$M_{k,n} = \exp\left(-j\frac{2\pi f x_k \sin(\theta_n)}{c}\right) , \tag{2.23}$$

where θ_n are the different constraint locations. The response vector, g, contains the look direction constraint (i.e., 1) in the first element, with the other constraint responses following. For instance, if three constraints were set (i.e., look direction plus two point constraints) and these additional point constraints were set to 3 dB down, g would be

$$g = \begin{bmatrix} 1.00 \\ 0.707 \\ 0.707 \end{bmatrix} . \tag{2.24}$$

Solution for Multiple Linear Constraints

The derivation of the solution is similar to that for the SP-ABF. Stated in mathematical terms, the problem is

$$\min_{w} w^H R w \quad \text{subject to } M^H w = g \text{ and } G_w \geq \delta^2 . \tag{2.25}$$

Forming the functional with the multiple linear equality constraints gives

$$F = w^H R w - \lambda\left(M^H w - g\right) \quad . \tag{2.26}$$

Taking the derivative with respect to the weight vector, and solving for λ, gives the solution for the weight vector,

$$w = R^{-1}M\left[M^H R^{-1} M\right]^{-1} g \quad . \tag{2.27}$$

This equation shows the solution without the white noise injected. Injecting white noise replaces R with $(R + \varepsilon I)$. Thus, the solution with the WNC included is

$$w = (R + \varepsilon I)^{-1} M\left[M^H (R + \varepsilon I)^{-1} M\right]^{-1} g \quad . \tag{2.28}$$

2.4.2 Implementation on the Alliant

The MLC-ABF has been implemented on the Alliant for the three types of constraint schemes described above (i.e., derivative, match-CBF, and point constraints). Only one type of constraint can be used at a time with the current code. As with the SP-ABF case, the CSM is decomposed into eigenvalues and eigenvectors using an SVD subroutine (Eq. (2.11)). Based on this decomposition and the solution for the weights, the expression for the inverse of the white noise gain (i.e., $1/G_W$) can be written explicitly as

$$\begin{aligned} w^H w = &\left\{\left(U diag[\sigma + \varepsilon]^{-1} U^H\right) M\left(M^H U diag[\sigma + \varepsilon]^{-1} U^H M\right)^{-1} g\right\}^H \times \\ &\left\{\left(U diag[\sigma + \varepsilon]^{-1} U^H\right) M\left(M^H U diag[\sigma + \varepsilon]^{-1} U^H M\right)^{-1} g\right\} \quad . \end{aligned} \tag{2.29}$$

This equation can be rewritten as

$$\begin{aligned} w^H w = &\left\{g^H\left[M^H U diag(\sigma + \varepsilon)^{-1} U^H M\right]^{-1} M^H U^H diag(\sigma + \varepsilon)^{-1} U\right\} \times \\ &\left\{U diag(\sigma + \varepsilon)^{-1} U^H M\left[M^H U diag(\sigma + \varepsilon)^{-1} U^H M\right]^{-1} g\right\} \quad . \end{aligned} \tag{2.30}$$

This equation is broken into components that can be computed once, and those changing with iteration. Two terms computed once and stored are

$$A = U^H M \text{ and } B_{ik}^j = \left(M^H U\right)_{ij} \left(U^H M\right)_{jk} \quad . \tag{2.31}$$

Consequently,

$$M^H U diag(\sigma + \varepsilon)^{-1} U^H M \tag{2.32}$$

is computed as

$$M^H U diag(\sigma + \varepsilon)^{-1} U^H M = B_{ik}^j \frac{1}{\left(\sigma_j + \varepsilon\right)} \quad . \tag{2.33}$$

Again, the iteration on white noise gain is performed if the solution without white noise injection does not satisfy the white noise gain constraint. Prior to iteration, the bounds of ε, which is parameterized as $\log(\varepsilon)$, are found using an initial bracketing routine adapted from a subroutine in *Numerical Recipes*.[4,*] After the initial bounds are set, a root bracketing function using Brent's method (i.e., the same as SP-ABF) is used to find the proper ε value, and the weights are calculated and stored for that frequency. After completing all frequencies in the CSM, the weights for that look direction are written to disk.

2.5 EXAMPLE BEAM PATTERNS AND SOLUTIONS

Figures 2.2 through 2.4 show examples of beam patterns calculated from the ABF weights for the different algorithms described above. All were calculated using the same four CSMs and the same array configurations. The CSMs were calculated from timeseries data obtained from a horizontal line array in a deep ocean environment. The time period between each CSM is approximately 15 s. In all cases, the white noise gain constraint was turned off (WNC = -100 dB). The range of sin θ in each figure is -2 to 2, thus showing the beam pattern in visible and invisible space. This allows inspection of the sidelobe structure. Also, mainlobe squinting at look directions near endfire can be seen more easily with this format.

---

* Subroutine ZBRAC, pp. 245-246.

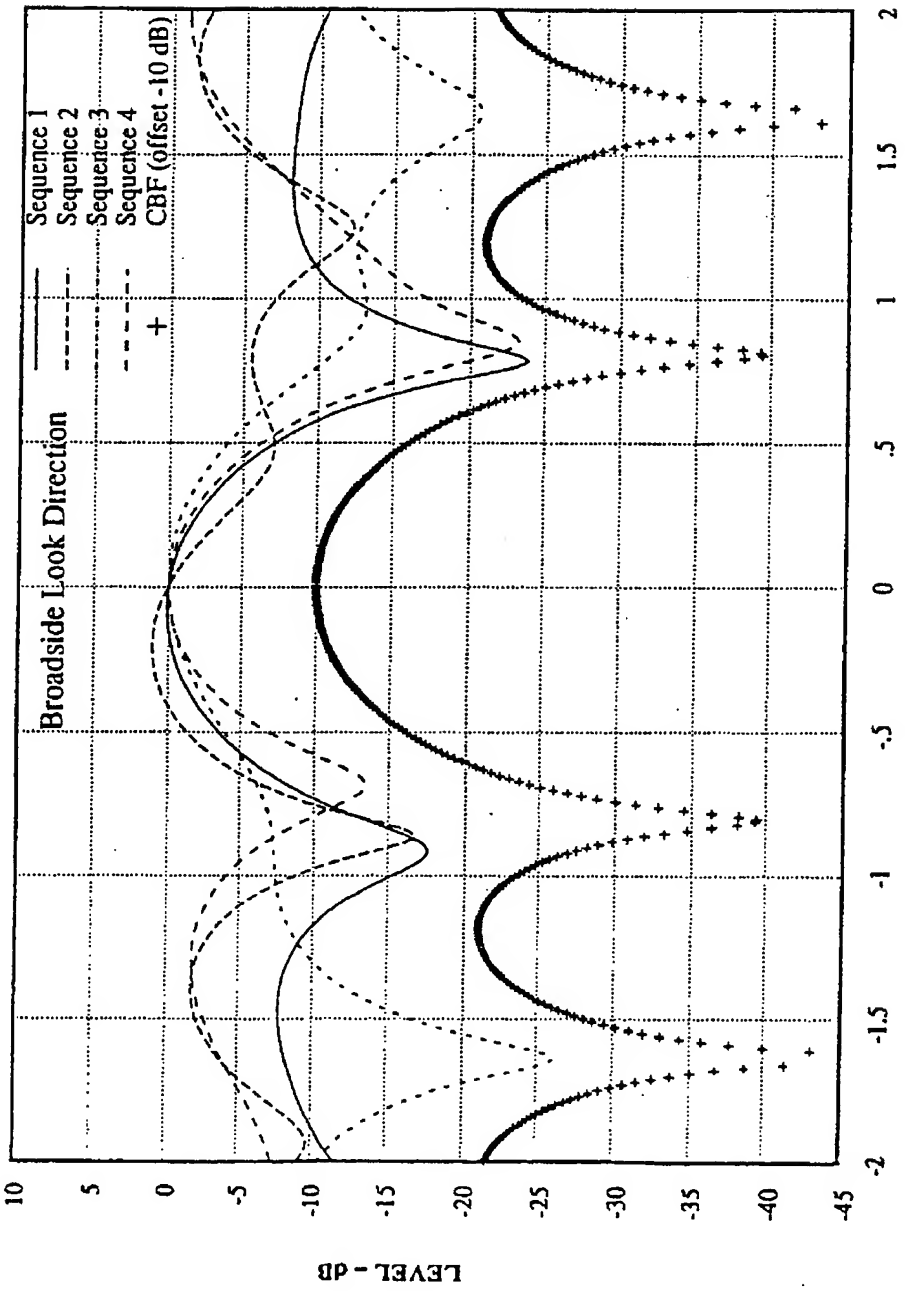

FIGURE 2.2
BEAM PATTERN FROM SP-ABF (WNC = -100),
FOUR HYDROPHONES, 60 Hz

ARL:UT
AS-92-115
RAG - DS
5 - 21 - 92

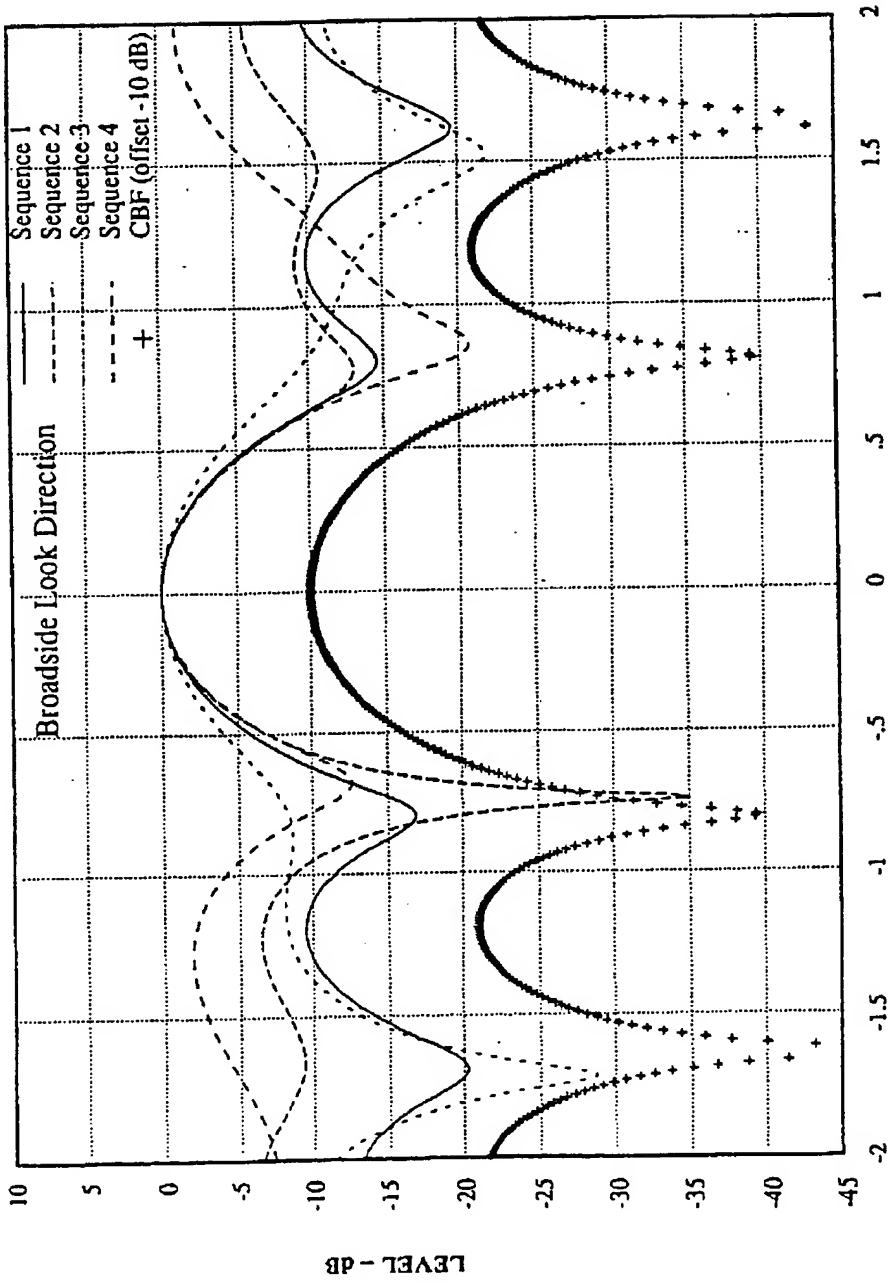

FIGURE 2.3
BEAM PATTERN FROM MLC-ABF (WNC = -100), 1st DERIVATIVE = 0,
FOUR HYDROPHONES, 60 Hz

ARL:UT
AS-92-116
RAG - DS
5 - 21 - 92

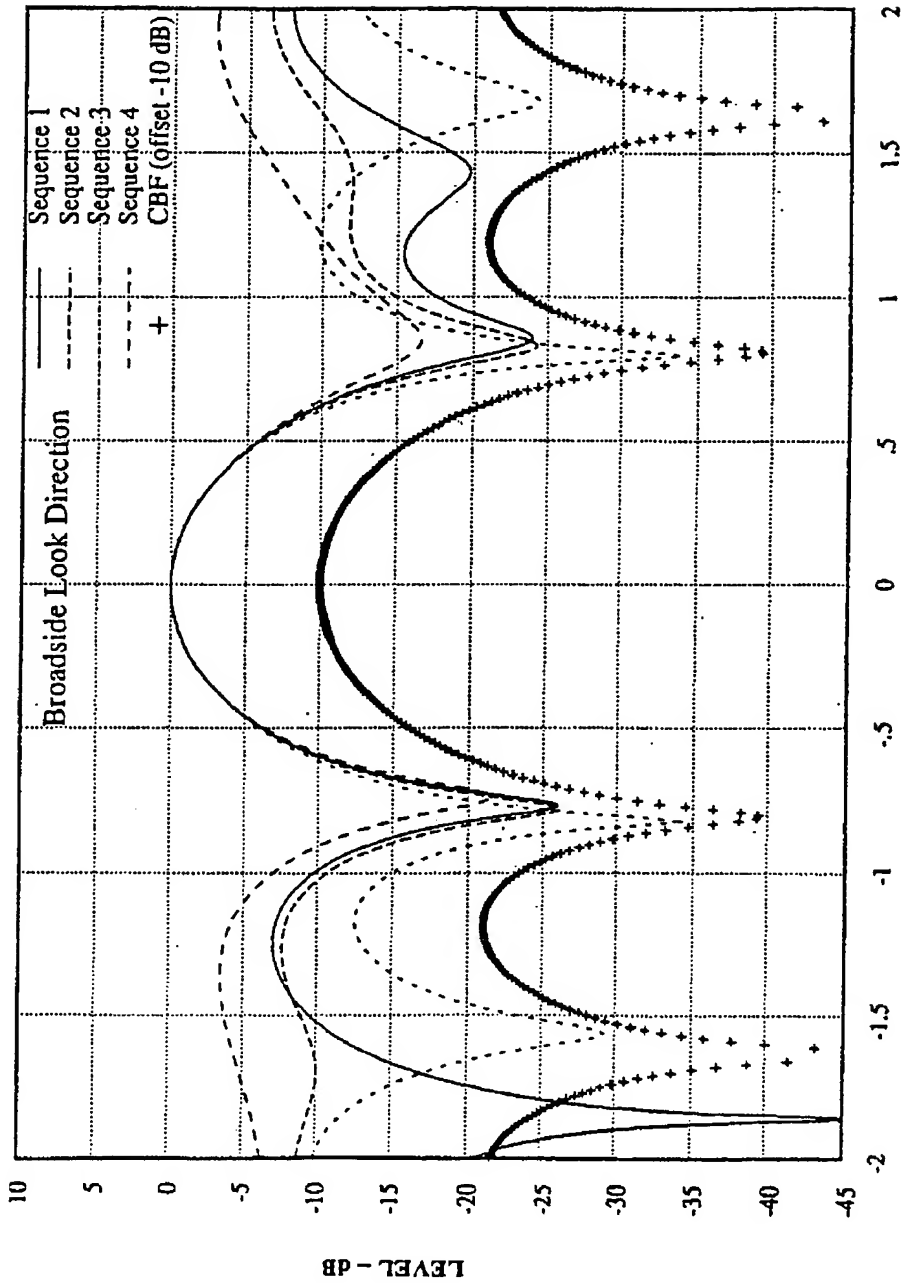

FIGURE 2.4
BEAM PATTERN FROM MLC-ABF (WNC = -100), MATCHED CBF AT -3 dB,
FOUR HYDROPHONES, 60 Hz

ARL:UT
AS-92-117
RAG - DS
5 - 21 - 92

Figure 2.2 shows the beam patterns based on the SP-ABF, and the equivalent CBF beam pattern. The CBF beam pattern is offset -10 dB for clarity. Clearly, each of the beam patterns satisfies the look direction constraint of unity gain (0 dB response). However, the slope of the response curves in the look direction, as well as the shapes of the mainlobes vary with the different input CSMs.

Figure 2.3 shows the beam pattern calculated from weights generated with the MLC-ABF program, with the first derivative set to zero in the look direction. The slope of each curve is zero at the look direction, and the response is much flatter (and closer to 0 dB) at locations near the look direction than in Figure 2.2.

Finally, Figure 2.4 shows the beam pattern calculated with weights generated with the MLC-ABF program using the Match-CBF option. In this case, a "three point" constraint was set with the two mainlobe maintenance constraints set to match the CBF response at the "3 dB down" points. Now, all of the response curves have essentially the same mainlobe shape between $\sin\theta = -0.4$ and 0.4.

3. SUMMARY AND CONCLUSIONS

Two adaptive beamforming (ABF) algorithms have been implemented at ARL:UT in order to assess the achievable performance of ABF. The single point (look direction) constrained (SP-ABF) and multiple linear constrained (MLC-ABF) adaptive beamformers were derived and explanations of their implementation were given in the previous sections. A discussion of the white noise gain constraint and its implementation was also included. Additional work examining the performance of these algorithms is under way using data obtained with ARL:UT acoustic data acquisition systems, and will be reported in a later document.

REFERENCES

1. F. W. Machell, "Algorithms for Broadband Processing and Display," Applied Research Laboratories Technical Letter No. 90-8 (ARL-TL-EV-90-8), Applied Research Laboratories, The University of Texas at Austin, March 1990.

2. H. Cox, "Resolving Power and Sensitivity to Mismatch of Optimum Array Processors," J. Acoust. Soc. Am. 54(3), 771-785 (1973).

3. H. Cox, R. M. Zeskind, and M. M. Owen, "Robust Adaptive Beamforming," IEEE Trans. Acoustics, Speech, and Signal Processing ASSP-35 (10), 1365-1376 (1987).

4. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, New York, 1986).

Should the Examiner have any questions regarding this Amendment, or the application in general, he is cordially invited to contact the undersigned at the number listed below.

Dated: October 28, 2005

Respectfully submitted,

By Sally A Ferrett

Sally A. Ferrett
Registration No.: 46,325
US NAVAL RESEARCH LABORATORY
4555 Overlook Ave, SW
Washington, DC 20375
(202) 767-3427
Attorney for Applicant